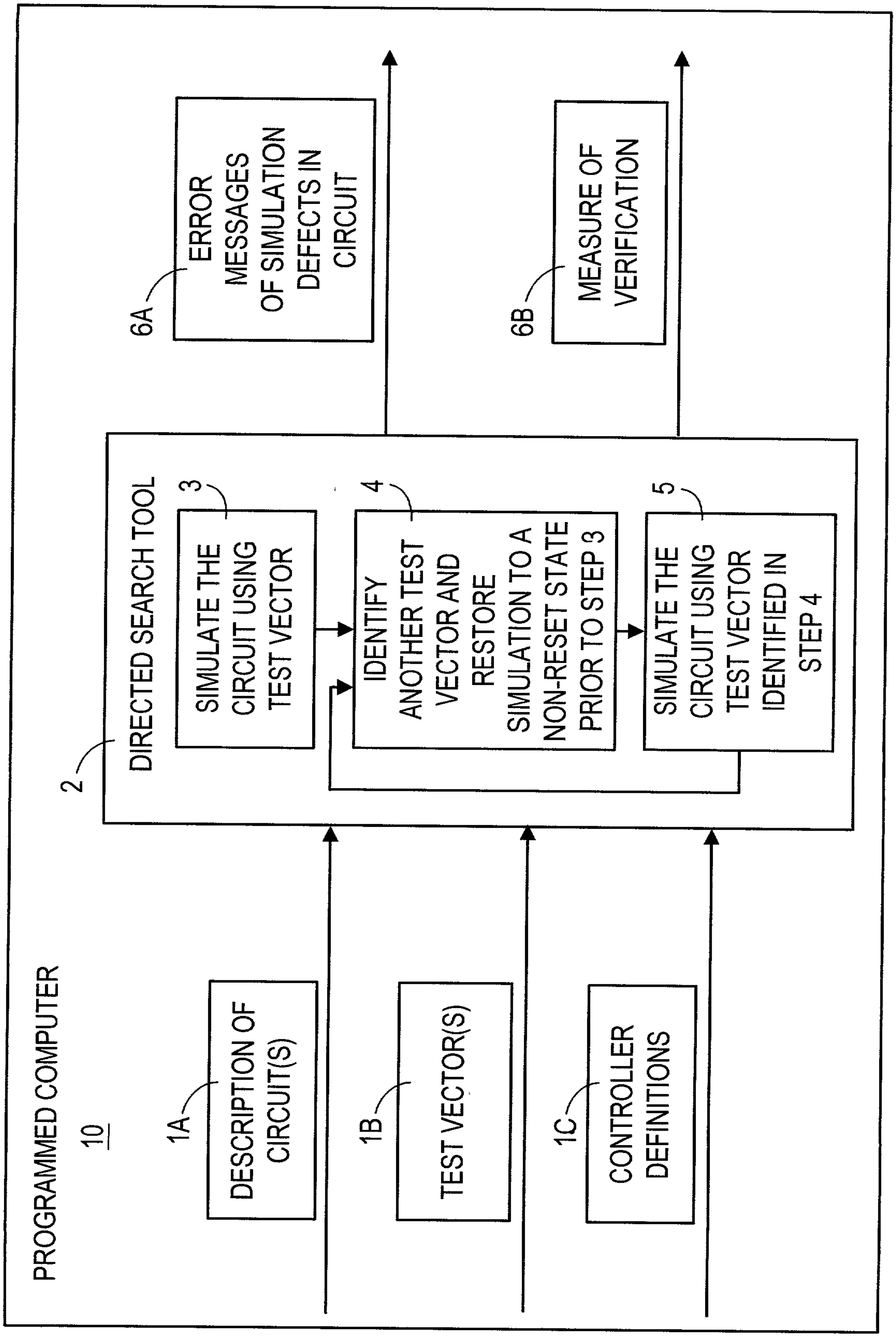

PROGRAMMED COMPUTER
10
1A
DESCRIPTION OF CIRCUIT(S)
1B
TEST VECTOR(S)
1C
CONTROLLER DEFINITIONS
2
DIRECTED SEARCH TOOL
3
SIMULATE THE CIRCUIT USING TEST VECTOR
4
IDENTIFY ANOTHER TEST VECTOR AND RESTORE SIMULATION TO A NON-RESET STATE PRIOR TO STEP 3
5
SIMULATE THE CIRCUIT USING TEST VECTOR IDENTIFIED IN STEP 4
6A
ERROR MESSAGES OF SIMULATION DEFECTS IN CIRCUIT
6B
MEASURE OF VERIFICATION

Fig. 1A

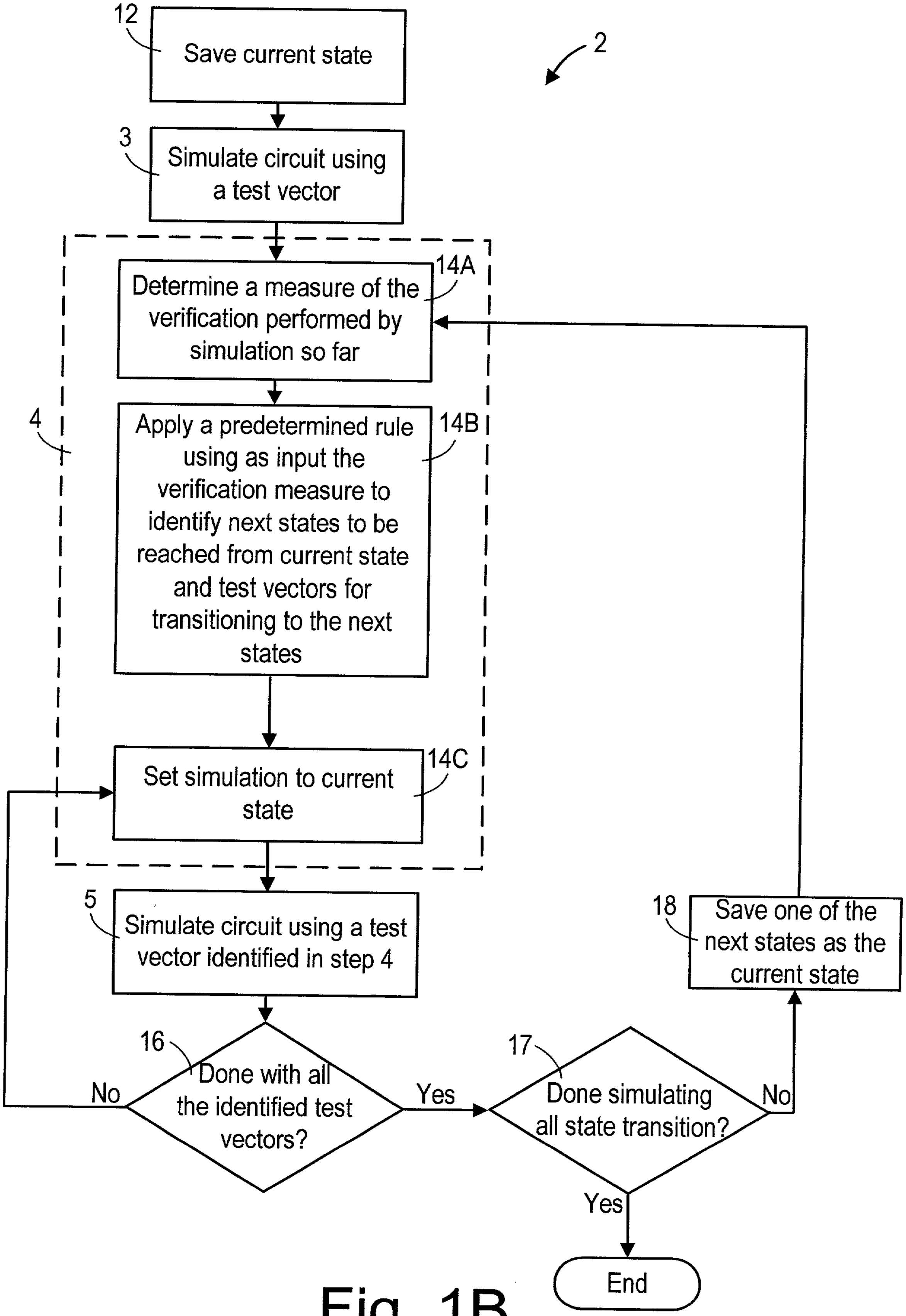

2
12
Save current state
3
Simulate circuit using a test vector
4
14A
Determine a measure of the verification performed by simulation so far
14B
Apply a predetermined rule using as input the verification measure to identify next states to be reached from current state and test vectors for transitioning to the next states
14C
Set simulation to current state
5
Simulate circuit using a test vector identified in step 4
16
Done with all the identified test vectors?
No
Yes
17
Done simulating all state transition?
No
Yes
18
Save one of the next states as the current state
End

Fig. 1B

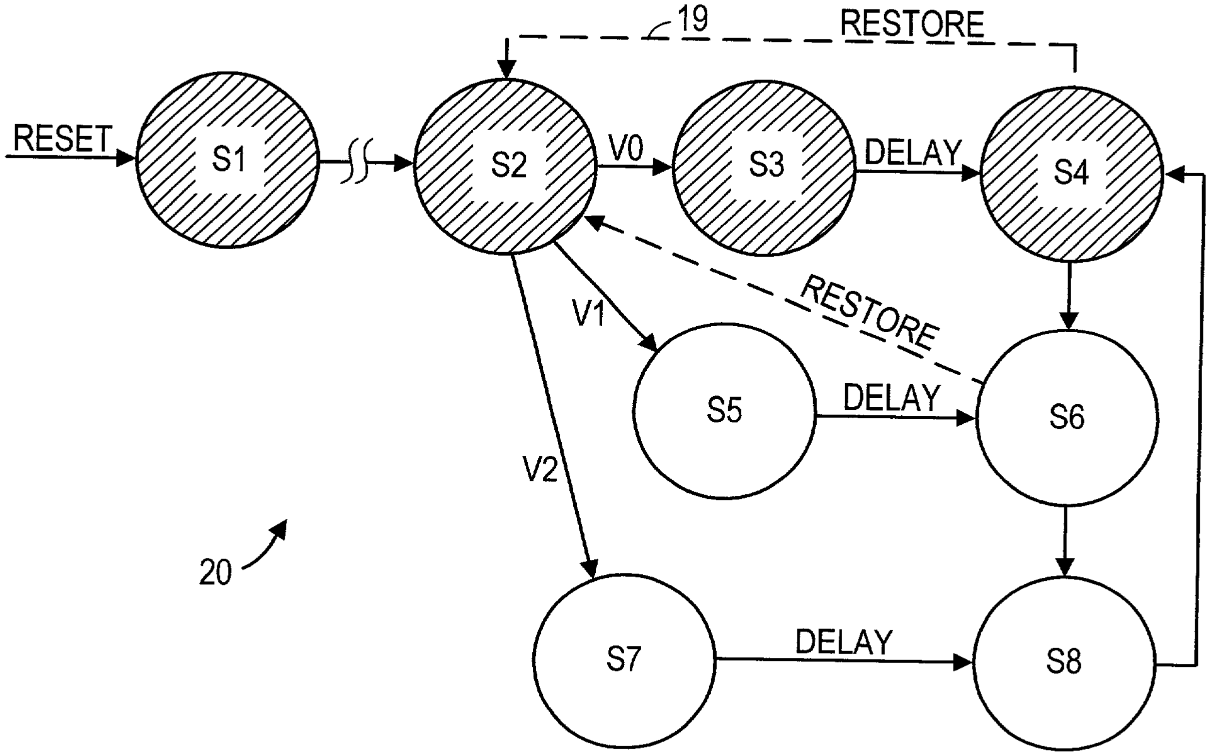

19
RESTORE
RESET
S1
S2
V0
S3
DELAY
S4
RESTORE
V1
S5
DELAY
S6
V2
20
S7
DELAY
S8

Fig. 1C

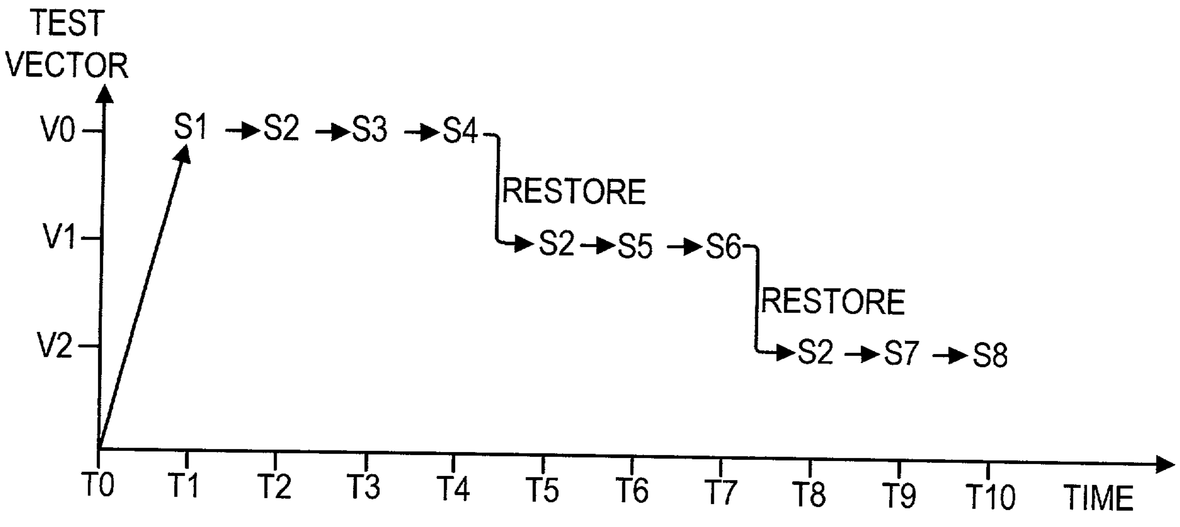

TEST VECTOR
V0
V1
V2
S1 → S2 → S3 → S4
RESTORE
→ S2 → S5 → S6
RESTORE
→ S2 → S7 → S8
T0
T1
T2
T3
T4
T5
T6
T7
T8
T9
T10
TIME

Fig. 1D

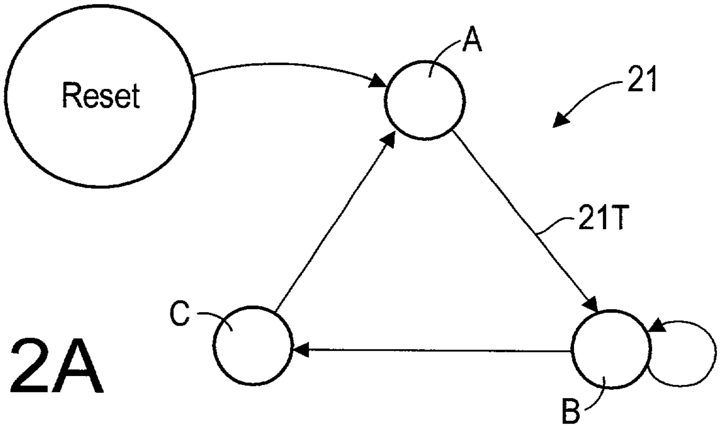


Fig. 2A

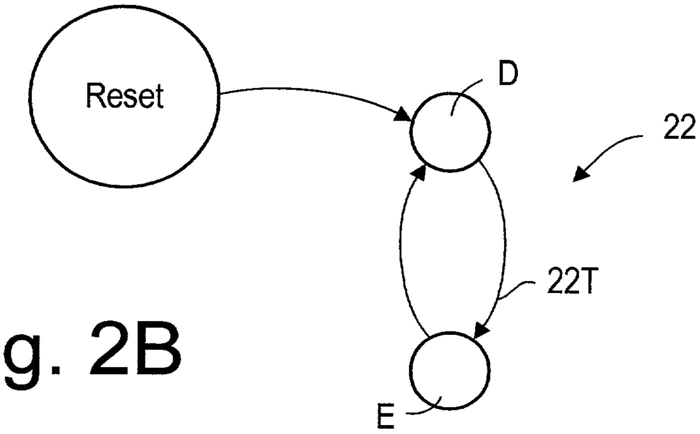


Fig. 2B

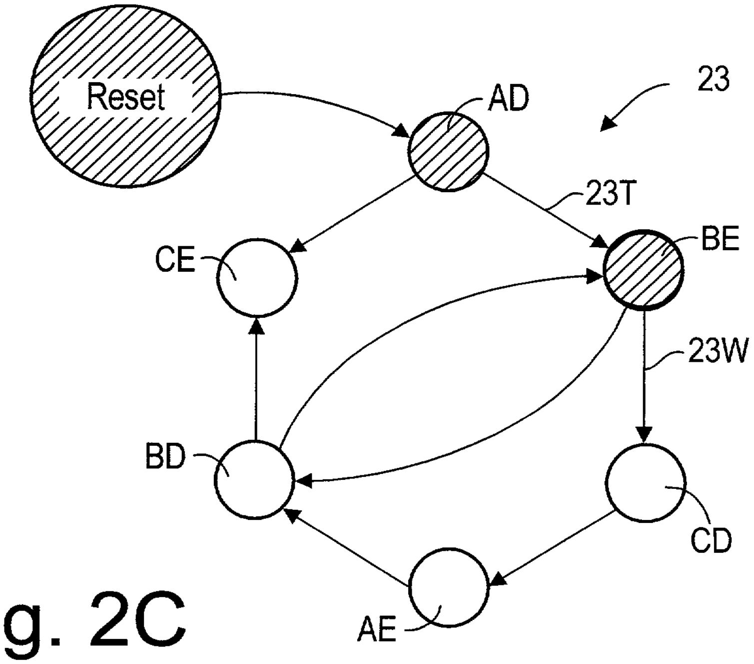


Fig. 2C

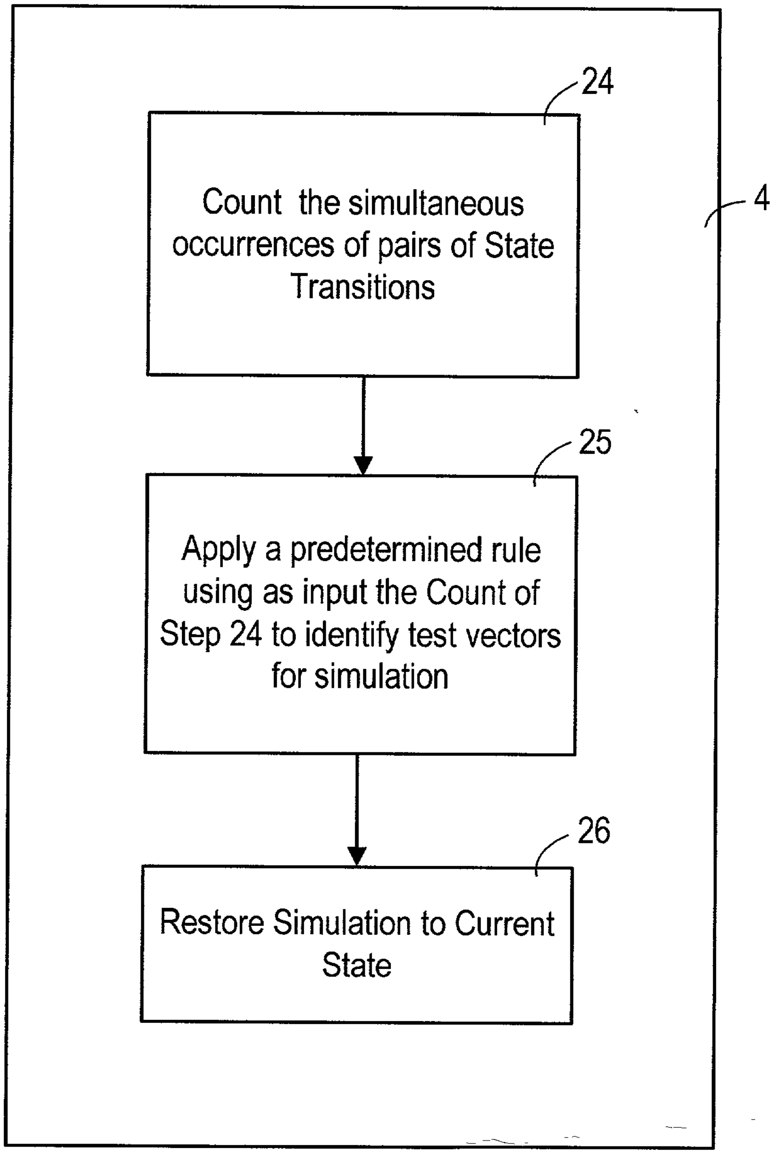

24
Count the simultaneous occurrences of pairs of State Transitions
4
25
Apply a predetermined rule using as input the Count of Step 24 to identify test vectors for simulation
26
Restore Simulation to Current State

Fig. 2D

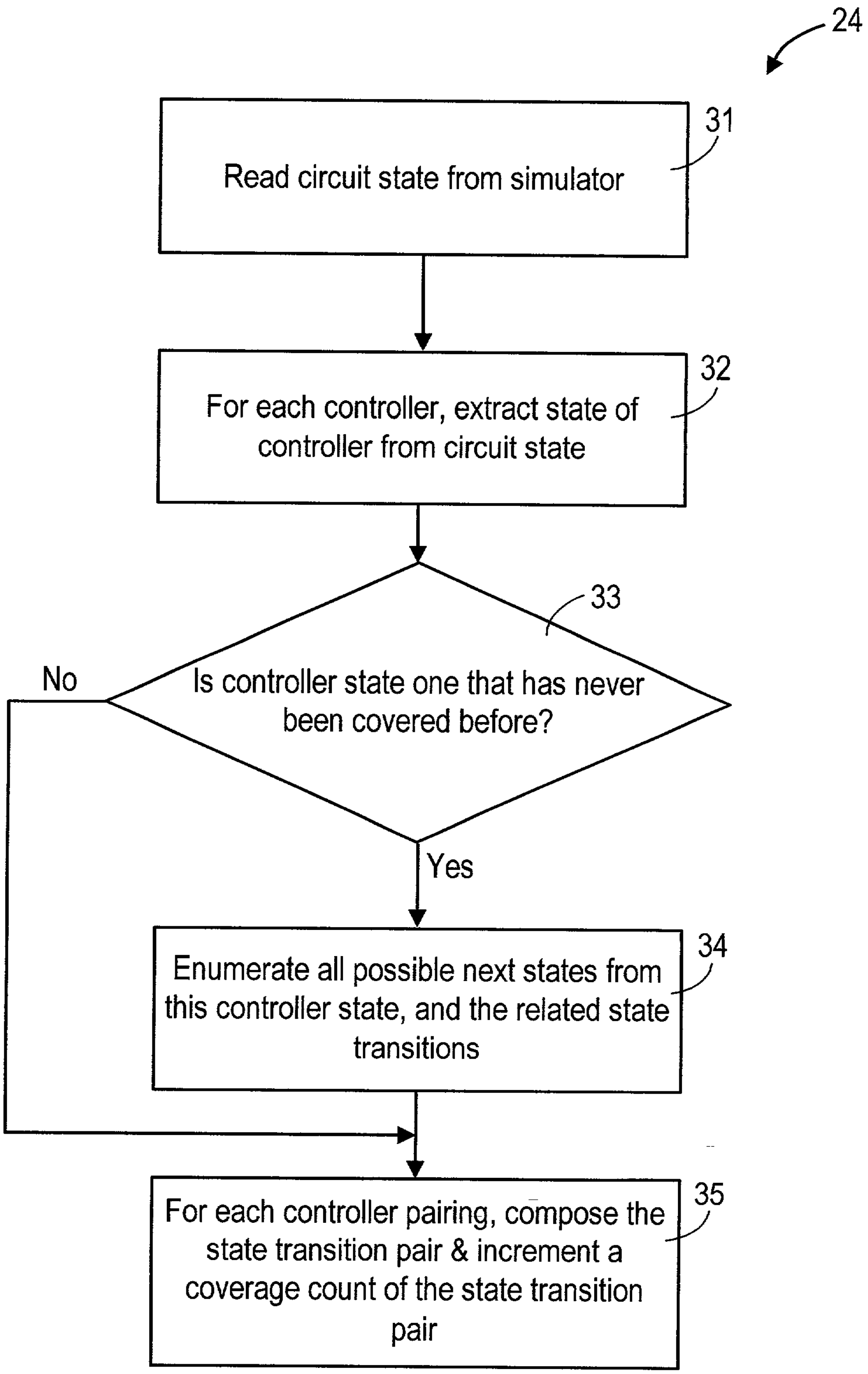

24
Read circuit state from simulator
31
For each controller, extract state of controller from circuit state
32
Is controller state one that has never been covered before?
33
No
Yes
Enumerate all possible next states from this controller state, and the related state transitions
34
For each controller pairing, compose the state transition pair & increment a coverage count of the state transition pair
35

Fig. 2E

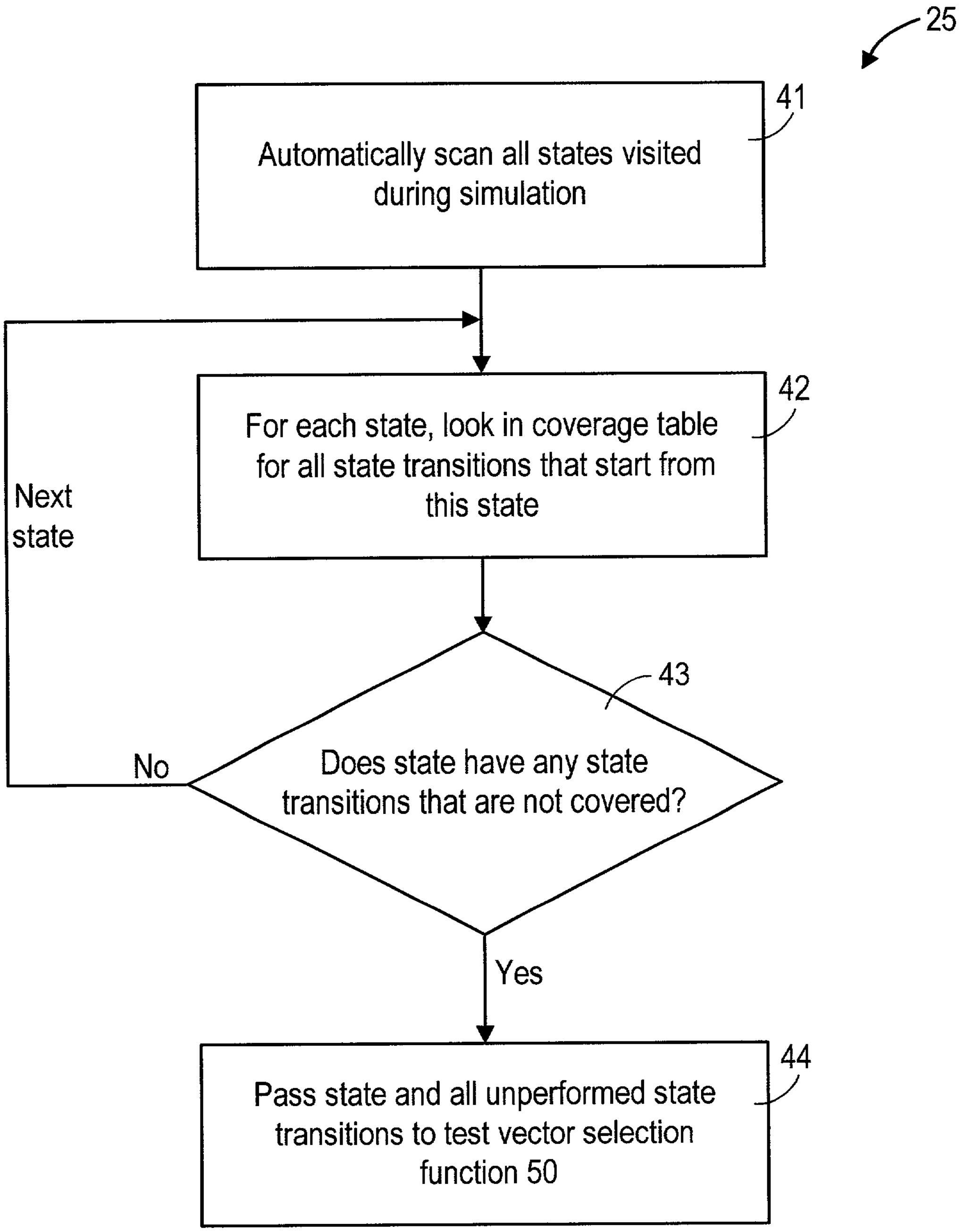

25
41
Automatically scan all states visited during simulation
42
For each state, look in coverage table for all state transitions that start from this state
Next state
43
Does state have any state transitions that are not covered?
No
Yes
44
Pass state and all unperformed state transitions to test vector selection function 50

Fig. 2F

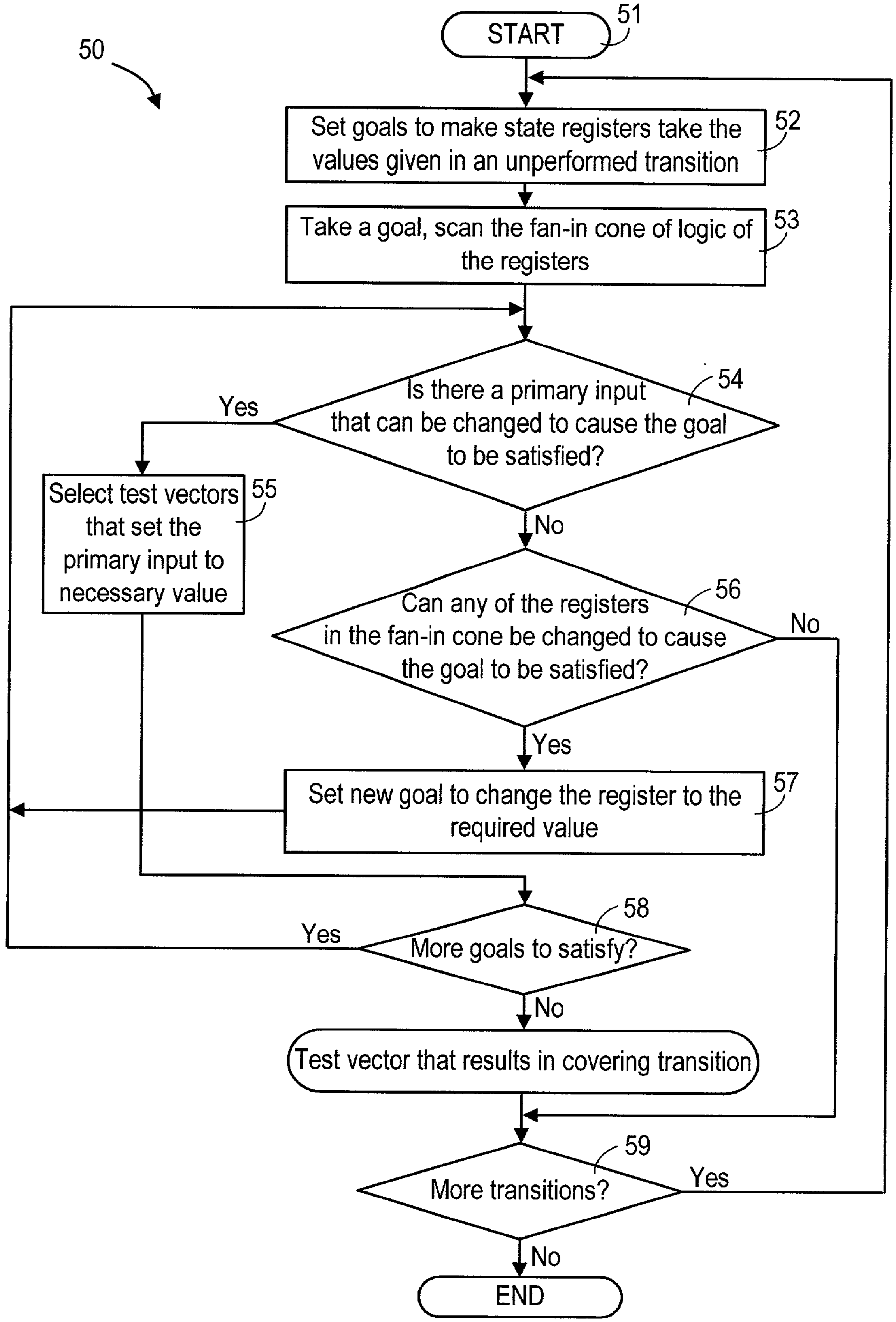

50
START
51
Set goals to make state registers take the values given in an unperformed transition
52
Take a goal, scan the fan-in cone of logic of the registers
53
Is there a primary input that can be changed to cause the goal to be satisfied?
54
Yes
Select test vectors that set the primary input to necessary value
55
No
Can any of the registers in the fan-in cone be changed to cause the goal to be satisfied?
56
No
Yes
Set new goal to change the register to the required value
57
More goals to satisfy?
58
Yes
No
Test vector that results in covering transition
More transitions?
59
Yes
No
END

Fig. 2G

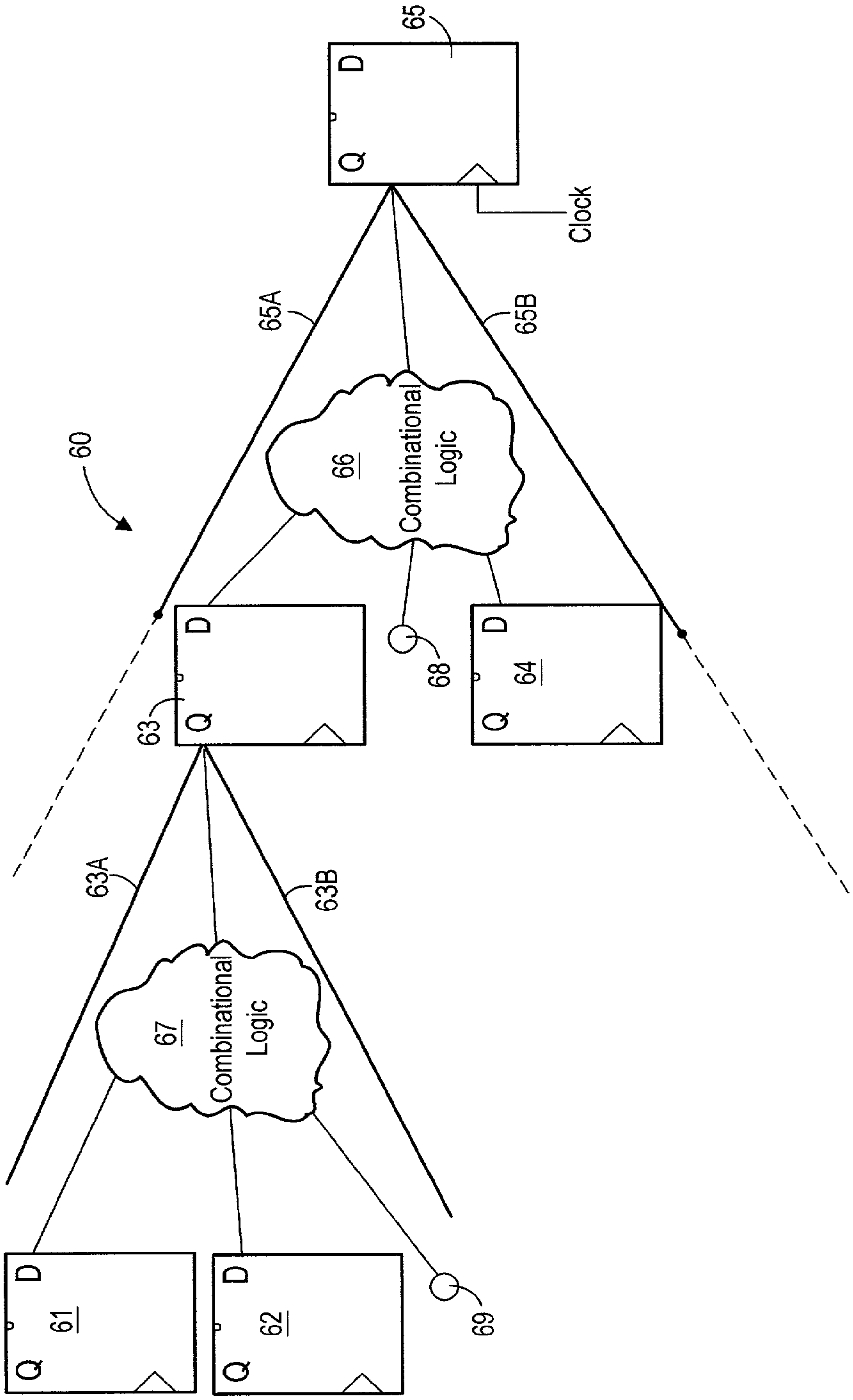

65
D
Q
Clock
65A
65B
60
66
Combinational Logic
63
D
Q
68
D
64
Q
63A
63B
67
Combinational Logic
D
61
Q
D
62
Q
69

Fig. 2H

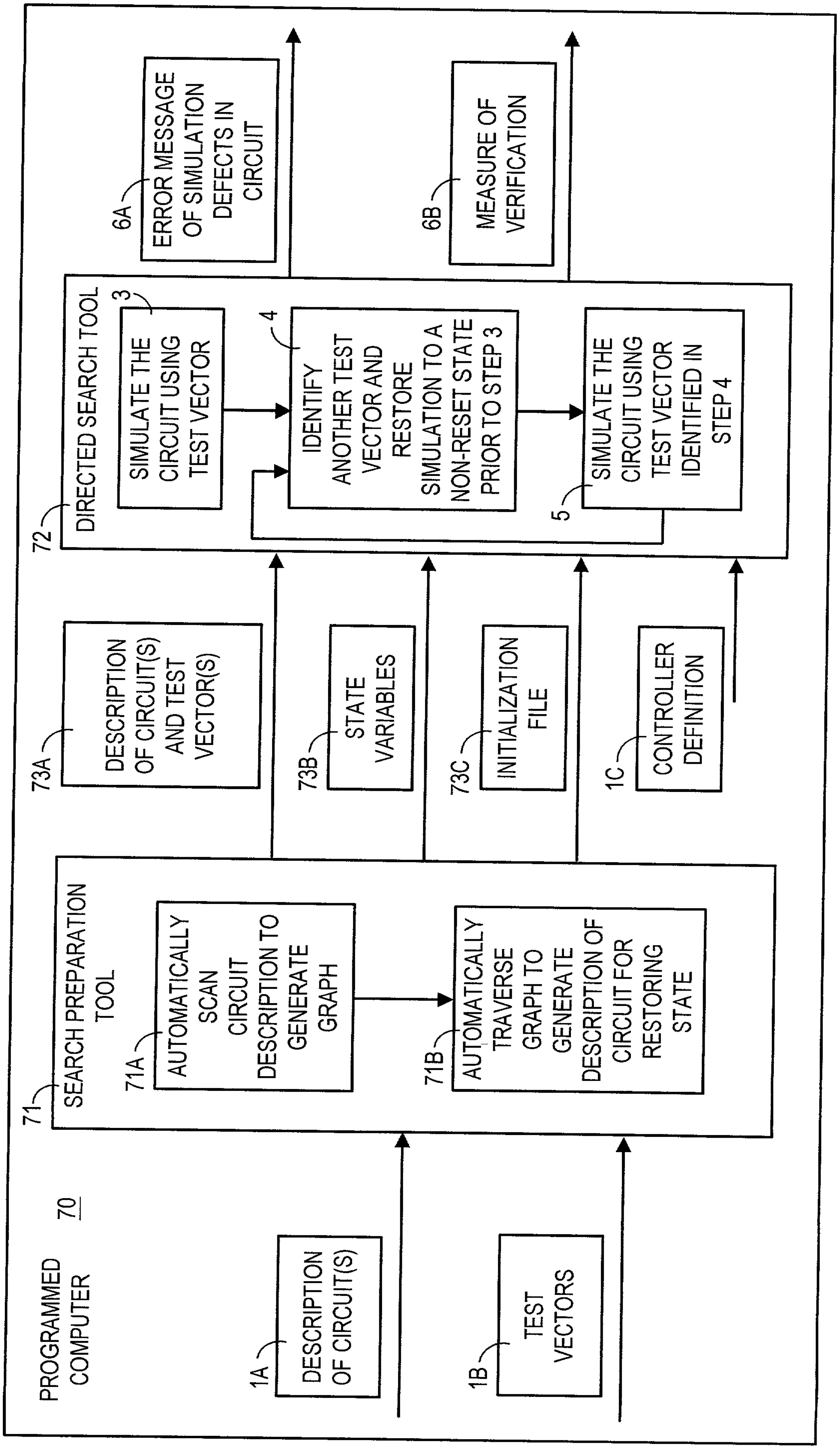

PROGRAMMED COMPUTER
70
1A
DESCRIPTION OF CIRCUIT(S)
1B
TEST VECTORS
71
SEARCH PREPARATION TOOL
71A
AUTOMATICALLY SCAN CIRCUIT DESCRIPTION TO GENERATE GRAPH
71B
AUTOMATICALLY TRAVERSE GRAPH TO GENERATE DESCRIPTION OF CIRCUIT FOR RESTORING STATE
73A
DESCRIPTION OF CIRCUIT(S) AND TEST VECTOR(S)
73B
STATE VARIABLES
73C
INITIALIZATION FILE
1C
CONTROLLER DEFINITION
72
DIRECTED SEARCH TOOL
3
SIMULATE THE CIRCUIT USING TEST VECTOR
4
IDENTIFY ANOTHER TEST VECTOR AND RESTORE SIMULATION TO A NON-RESET STATE PRIOR TO STEP 3
5
SIMULATE THE CIRCUIT USING TEST VECTOR IDENTIFIED IN STEP 4
6A
ERROR MESSAGE OF SIMULATION DEFECTS IN CIRCUIT
6B
MEASURE OF VERIFICATION

Fig. 3A

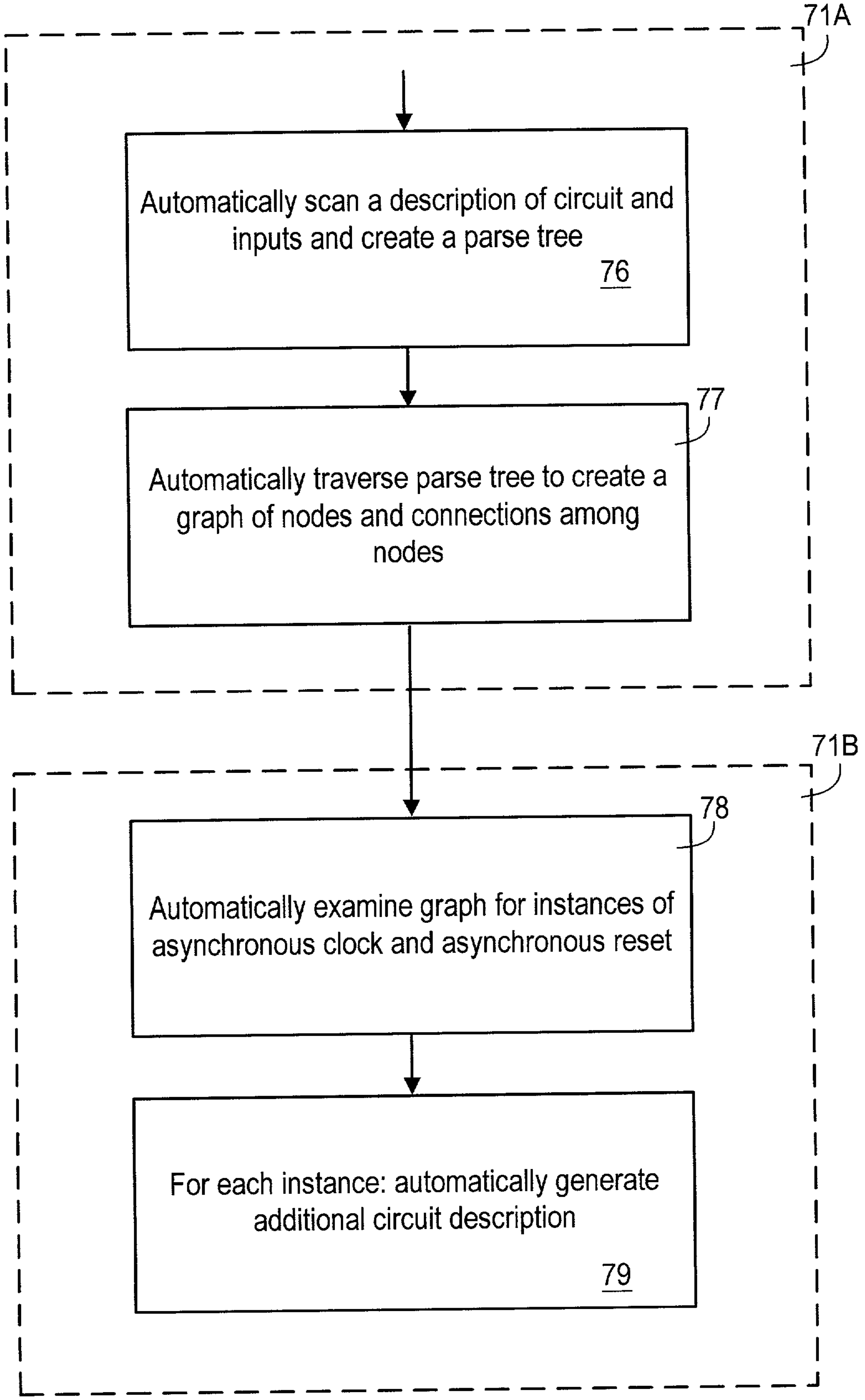

71A
Automatically scan a description of circuit and inputs and create a parse tree
76
77
Automatically traverse parse tree to create a graph of nodes and connections among nodes
71B
78
Automatically examine graph for instances of asynchronous clock and asynchronous reset
For each instance: automatically generate additional circuit description
79

Fig. 3B

Fig. 3C

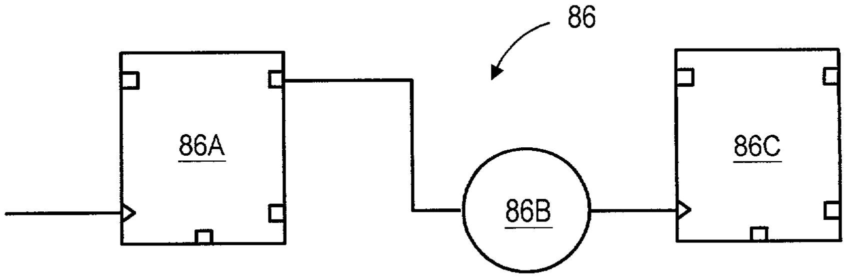


Fig. 3D

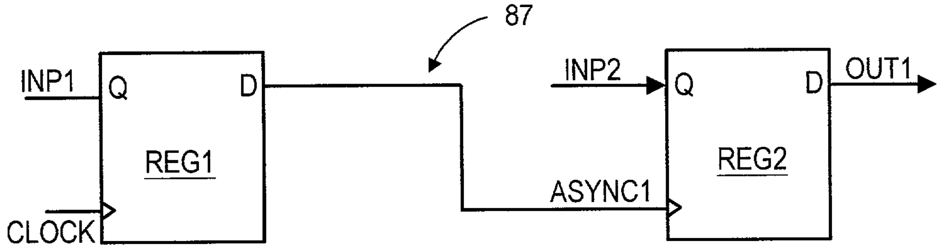


Fig. 3E

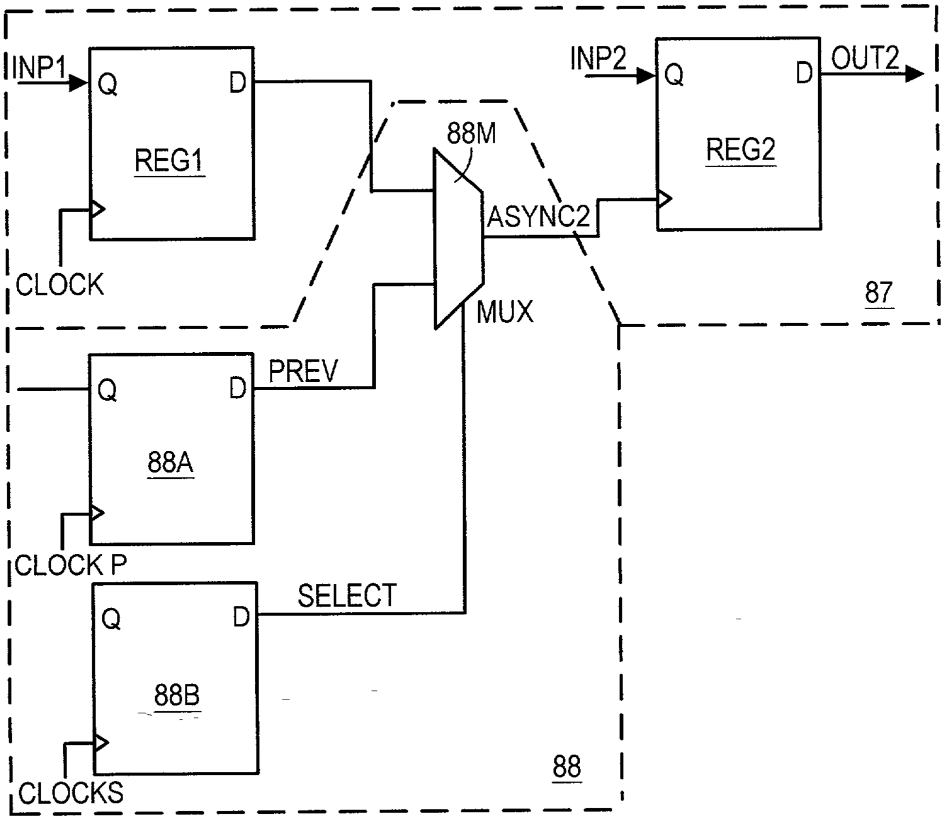
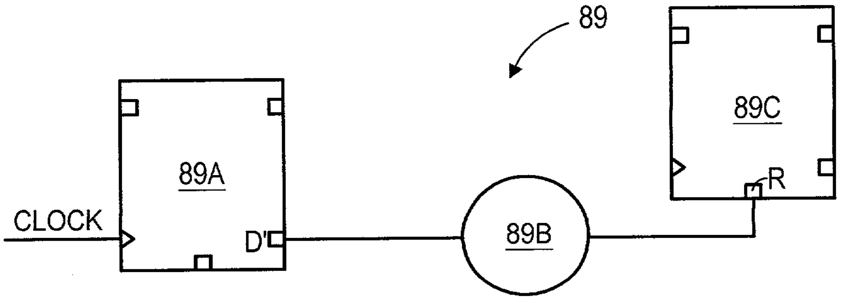
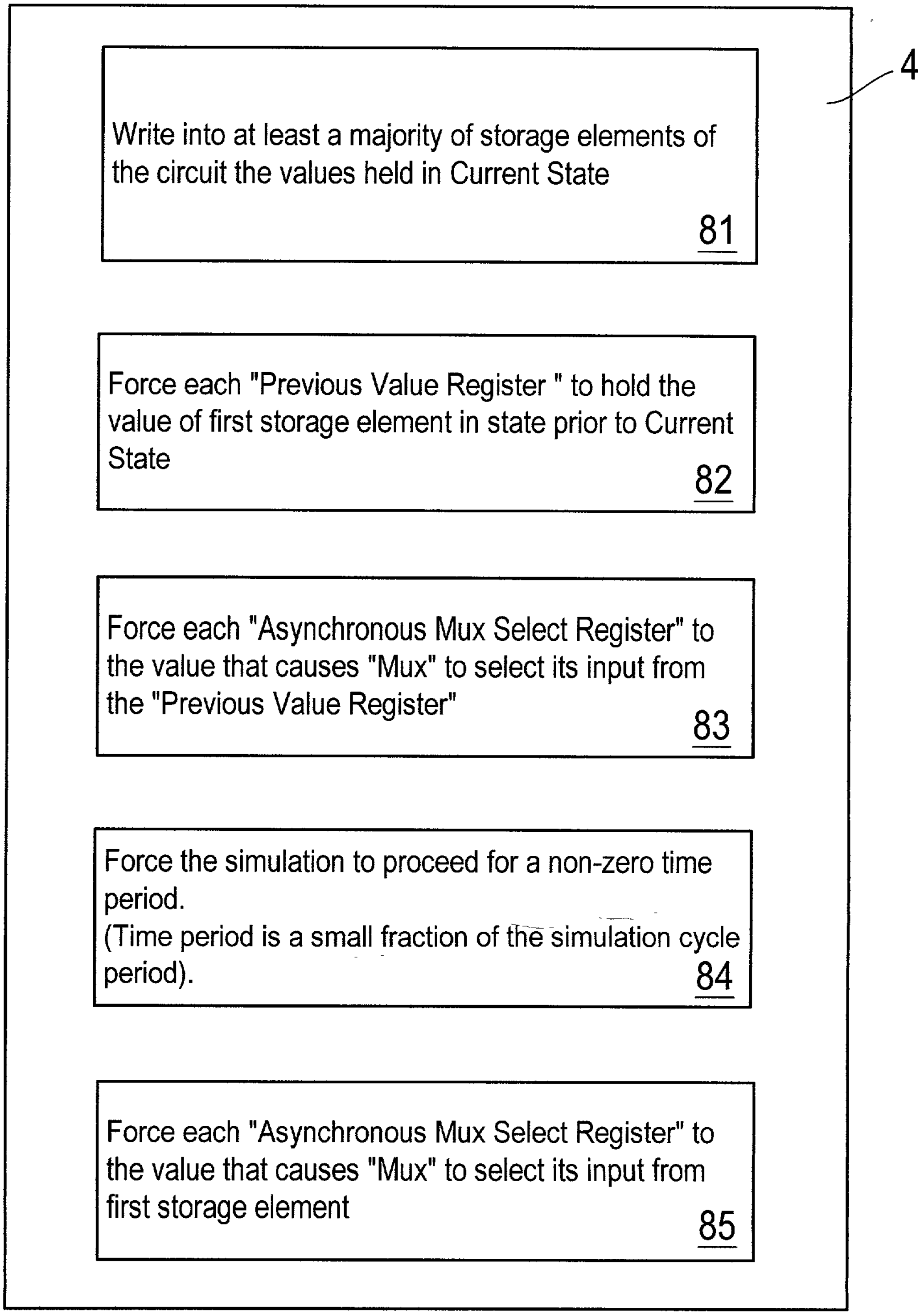


Fig. 3F





Write into at least a majority of storage elements of the circuit the values held in Current State 81

Force each "Previous Value Register " to hold the value of first storage element in state prior to Current State 82

Force each "Asynchronous Mux Select Register" to the value that causes "Mux" to select its input from the "Previous Value Register" 83

Force the simulation to proceed for a non-zero time period.
(Time period is a small fraction of the simulation cycle period). 84

Force each "Asynchronous Mux Select Register" to the value that causes "Mux" to select its input from first storage element 85

Fig. 3G

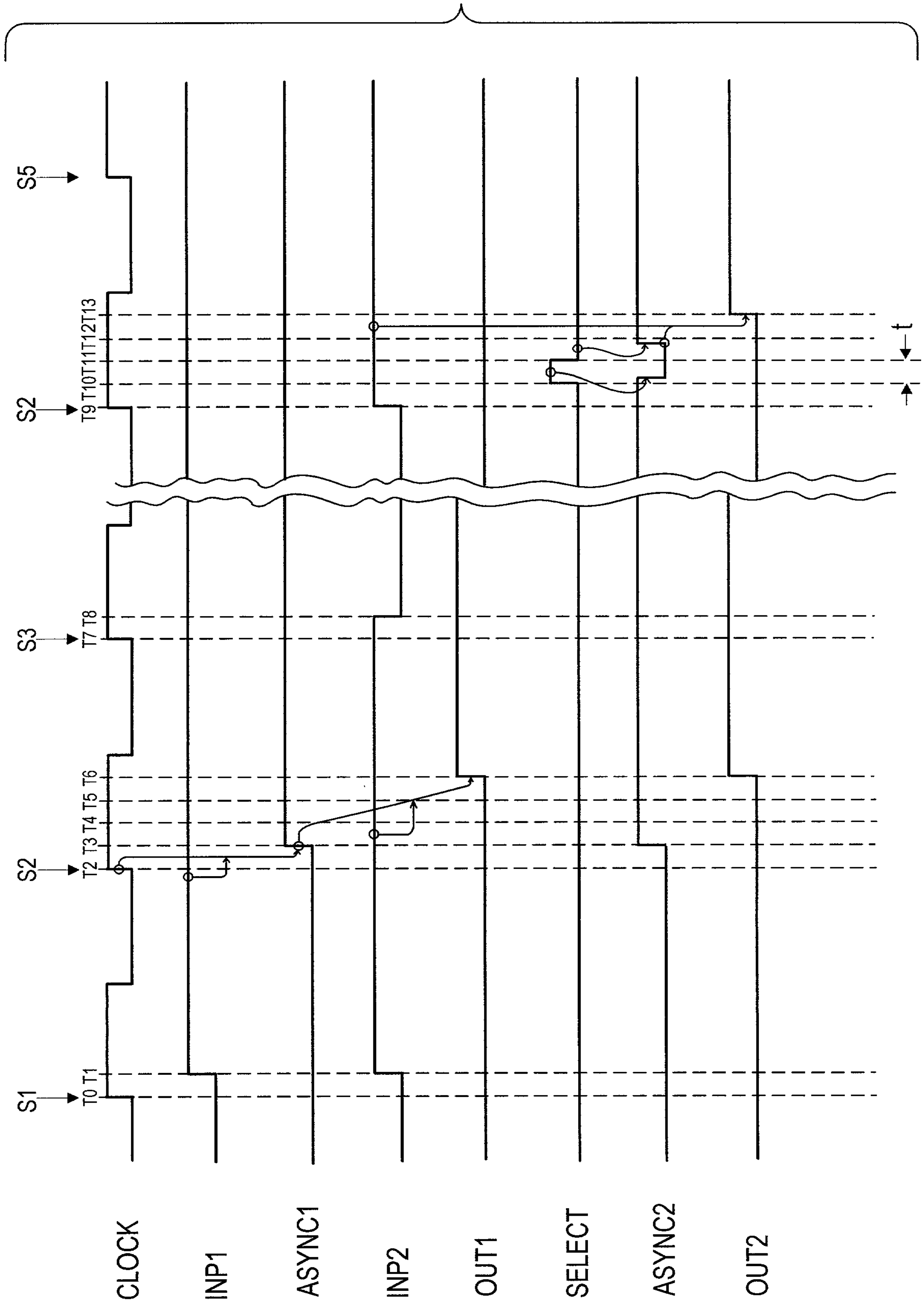

S1
T0 T1
S2
T2 T3 T4 T5 T6
S3
T7 T8
S2
T9 T10 T11 T12 T13
S5
t
CLOCK
INP1
ASYNC1
INP2
OUT1
SELECT
ASYNC2
OUT2

Fig. 3H

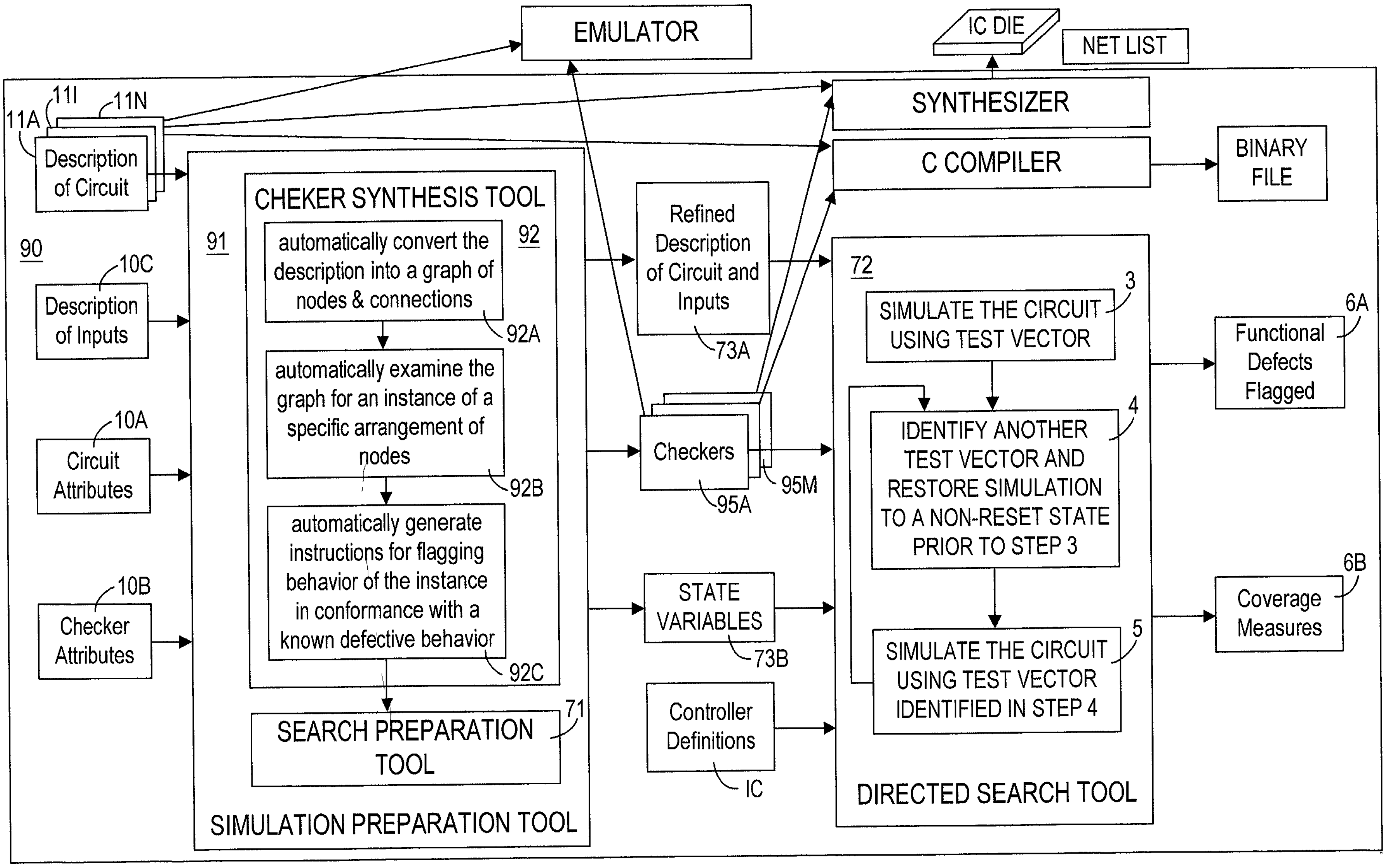

EMULATOR
IC DIE
NET LIST
SYNTHESIZER
C COMPILER
BINARY FILE
11A
11I
11N
Description of Circuit
90
10C
Description of Inputs
10A
Circuit Attributes
10B
Checker Attributes
91
CHEKER SYNTHESIS TOOL
92
automatically convert the description into a graph of nodes & connections
92A
automatically examine the graph for an instance of a specific arrangement of nodes
92B
automatically generate instructions for flagging behavior of the instance in conformance with a known defective behavior
92C
SEARCH PREPARATION TOOL
71
SIMULATION PREPARATION TOOL
Refined Description of Circuit and Inputs
73A
Checkers
95A
95M
STATE VARIABLES
73B
Controller Definitions
IC
72
SIMULATE THE CIRCUIT USING TEST VECTOR
3
IDENTIFY ANOTHER TEST VECTOR AND RESTORE SIMULATION TO A NON-RESET STATE PRIOR TO STEP 3
4
SIMULATE THE CIRCUIT USING TEST VECTOR IDENTIFIED IN STEP 4
5
DIRECTED SEARCH TOOL
Functional Defects Flagged
6A
Coverage Measures
6B

Fig. 4